IN THE CLAIMS:

1. (Currently Amended) A method comprising:

detecting a first splice indicator using transport packet demultiplexer hardware, wherein detecting the first splice indicator includes generating a first splice interrupt based upon the first splice indicator;

determining a new packet identifier when, in response to detecting the first splice indicator, it is determined that a first splice state has been encountered, wherein the first splice state is based upon a first splice countdown value parsed by the transport packet demultiplexer hardware, wherein the new packet identifier is determined in response to the first splice interrupt;

detecting a second splice indicator using the transport packet demultiplexer hardware, wherein detecting the second splice indicator includes generating a second splice interrupt based upon the second splice indicator; and

using the new packet identifier in response to the second splice interrupt.

2. (Previously Presented) The method of claim 1 further comprising:

loading the new packet identifier into a shadow register after determining the new packet identifier and before using the new packet identifier.

3. (Previously Presented) The method of claim 2, wherein using the new packet identifier further comprises loading the contents of the shadow register into a main register.

4. (Previously Presented) The method of claim 2, wherein using the new packet identifier further comprises using the shadow register as the main register.

5. (Previously Presented) The method of claim 1, wherein detecting the first splice indicator includes using an adaptation field parser portion of the transport packet demultiplexer hardware.

6. (Canceled)

7. (Canceled)

8. (Original) The method of claim 7, wherein the first splice indicator and the second splice indicator represent different occurrences of a common event.

9. (Original) The method of claim 8, wherein the common event is the assertion of a splice flag.

10. (Canceled)

11. (Previously Presented) The method of claim 1, wherein determining further includes the first splice countdown value being a positive value.

12. (Previously Presented) The method of claim 1 further including:

using the new packet identifier in response to the second splice interrupt, when, in response to detecting the second splice indicator it is determined that a second splice state has been encountered, wherein the second splice state is based upon a second splice countdown value parsed by the transport packet demultiplexer hardware.

13. (Previously Presented) The method of claim 12, wherein using further includes the second splice countdown value being a zero value.

14. (Previously Presented) The method of claim 13, wherein determining further includes the first splice countdown value being a positive value.

15. (Previously Presented) The method of claim 1, further comprising:

detecting a third splice indicator using transport packet demultiplexer hardware;

requesting acquisition of a current program management table in response to the third splice indicator.

16. (Currently Amended) A method comprising:

detecting a first splice indicator using transport packet demultiplexer hardware;

determining a new packet identifier when, in response to detecting the first splice indicator, it is determined that a first splice state has been encountered, wherein the first splice state is based upon a first splice countdown value parsed by the transport packet demultiplexer hardware;

detecting a second splice indicator using the transport packet demultiplexer hardware;

using the new packet identifier in response to the second splice indicator;[[.]]

detecting a third splice indicator using transport packet demultiplexer hardware; and

requesting acquisition of a current program management table when, in response to detecting the third splice indicator, it is determined that a third splice state has been encountered, wherein the third splice state is based upon a third splice countdown value parsed by the transport packet demultiplexer hardware.

17. (Previously Presented) The method of claim 16, wherein determining further includes the first splice countdown value being a negative value.

18. (Previously Presented) The method of claim 16, further comprising verifying the new packet identifier.

19. (Currently Amended) The method of claim 1, wherein:

using the new packet identifier in response to the second splice interrupt further includes using the new packet identifier in response to the second splice interrupt when the new packet identifier is associated with a first program type.

20. (Currently Amended) The method of claim 19, wherein the first program type is mutually exclusive from a second program type, and the second program type is a commercial[[s]] type.

21. (Previously Presented) A method comprising:
detecting a first splice indicator using transport packet demultiplexer hardware, wherein detecting the first splice indicator includes generating a first splice interrupt;
determining a new packet identifier in response to the first splice interrupt;
detecting a second splice indicator using the transport packet demultiplexer hardware;
using the new packet identifier in response to the second splice indicator;
detecting a third splice indicator using transport packet demultiplexer hardware; and
requesting acquisition of a current program management table in response to the third splice indicator.

22. (Previously Presented) The method of claim 21 further comprising:
loading the new packet identifier into a shadow register after determining the new packet identifier and before using the new packet identifier.

23. (Previously Presented) The method of claim 22, wherein using the new packet identifier further comprises loading the contents of the shadow register into a main register.

24. (Previously Presented) The method of claim 22, wherein using the new packet identifier further comprises using the shadow register as the main register.

25. (Previously Presented) The method of claim 21, wherein detecting the first splice indicator includes using an adaptation field parser portion of the transport packet demultiplexer hardware.

26. (Canceled)

27. (Previously Presented) The method of claim 21, wherein
detecting the second splice indicator includes generating a second splice interrupt based upon the second splice indicator; and
using the new packet identifier occurs in response to the second splice interrupt.

28. (Previously Presented) The method of claim 27, wherein the first splice indicator and the second splice indicator represent different occurrences of a common event.

29. (Previously Presented) The method of claim 28, wherein the common event is the assertion of a splice flag.

30. (Previously Presented) A method comprising:
detecting a first splice indicator using transport packet demultiplexer hardware;
determining a new packet identifier in response to the first splice indicator;
detecting a second splice indicator using the transport packet demultiplexer hardware;
using the new packet identifier in response to the second splice indicator;
detecting a third splice indicator using transport packet demultiplexer hardware; and
requesting acquisition of a current program management table when, in response to detecting the third splice indicator, it is determined that a third splice state has been encountered, wherein the third splice state is based upon a third splice countdown value parsed by the transport packet demultiplexer hardware.

31. (Previously Presented) The method of claim 30, wherein determining further includes the third splice countdown value being a negative value.

32. (Previously Presented) The method of claim 30, further comprising verifying the new packet identifier.

33. (Currently Amended) The method of claim 30, wherein:
using the new packet identifier in response to the second splice indicator further includes using the new packet identifier in response to the second splice indicator when the new packet identifier is associated with a first program type.

34. (Previously Presented) The method of claim 33, wherein the first program type is mutually exclusive from a second program type, and the second program type is a commercial type.

35. (Previously Presented) A system comprising:

a means for detecting a first splice indicator using transport packet demultiplexer hardware, wherein the means for detecting the first splice indicator includes a means for generating a first splice interrupt in response to the first splice indicator;

a means for determining a new packet identifier when, in response to detecting the first splice indicator it is determined that a first splice state has been encountered, wherein the first splice state is based upon a first splice countdown value parsed by the transport packet demultiplexer hardware, wherein the new packet identifier is determined in response to the first splice interrupt;

a means for detecting a second splice indicator using the transport packet demultiplexer hardware, wherein the means for detecting the second splice indicator includes a means for generating a second splice interrupt in response to the first splice indicator; and

a means for using the new packet identifier in response to the second splice interrupt.

36. (Previously Presented) The system of claim 35, further comprising:

a means for loading the new packet identifier into a shadow register after determining the new packet identifier and before using the new packet identifier.

37. (Canceled)

38. (Canceled)

39. (Currently Amended) A method comprising:

detecting a first splice indicator using transport packet demultiplexer hardware; and

requesting acquisition of a current program management table when, in response to detecting the first splice indicator it is determined that a first splice state has been encountered, wherein the first splice state is based upon a first splice countdown value parsed by the transport packet demultiplexer hardware.